Amendments to the Claims

No Claims are being amended by the present Response. A listing of all currently pending claims is represented below.

1 – 10. (Canceled)

11.   (Previously Presented) A computer-implemented method comprising:

generating a transformation file by employing a query language, said transformation file containing a set of rules to transform data between two or more formats having different shapes;

attaching the transformation file to a workflow, such that the set of rules are referenced from inside the workflow;

associating, at compile time, a first shape of a first data structure with an intermediate shape representation based on the set of rules of the transformation file, wherein the first shape defines a structure and layout of data in the first data structure;

receiving a second data structure during runtime execution of said workflow, said second data structure having a second shape that is different from the first shape of the first data structure;

applying the intermediate shape representation to the second data structure;

mapping the second data structure from the intermediate shape representation to the first shape of the first data structure; and

generating a runtime object containing the data obtained from the second data structure and having the first shape of the first data structure and using the runtime object as input for a component of said workflow.

12 –18.   (Canceled)

19.   (Previously Presented)   A computer-readable storage medium carrying one or more sequences of instructions, which when executed by one or more processors, cause the one or more processors to:

generate a transformation file by employing a query language, said transformation file containing a set of rules to transform data between two or more formats having different shapes;

attach the transformation file to a workflow, such that the set of rules are referenced from inside the workflow;

associate, at compile time, a first shape of a first data structure with an intermediate shape representation based on the set of rules of the transformation file, wherein the first shape defines a structure and layout of data in the first data structure;

receive a second data structure during runtime execution of said workflow, said second data structure having a second shape that is different from the first shape of the first data structure;

apply the intermediate shape representation to the second data structure;

map the second data structure from the intermediate shape representation to the first shape of the first data structure; and

generate a runtime object containing the data obtained from the second data structure and having the first shape of the first data structure and using the runtime object as input for a component of said workflow.

20 – 22. (Canceled)

23.   (Previously Presented) The computer-implemented method of claim 1, further comprising:

compiling the transformation file to generate a compiled plan; and

storing the compiled plan for use at runtime, such that the compiled plan is used to convert data from the intermediate representation.

24.   (Previously Presented) The computer-implemented method of claim 1, wherein multiple data structure sources are combined into a single result object by using the intermediate shape representation.

25.   (Previously Presented)   The computer-implemented method of claim 1, wherein the transformation file is included in the workflow by implementing a control within the workflow, said control invoking transformations during send or receive operations from the workflow.

26.   (Previously Presented) A method for declaratively transforming data between different data formats, said method comprising:

employing a query language to associate a default extensible markup language (XML) representation with a data shape of each JAVA class in a set of JAVA classes, wherein the set of JAVA classes are used to communicate information in a workflow;

invoking the workflow;

receiving an XML document containing data as part of execution of said workflow, said data in the XML document having an XML data shape;

applying the default XML representation to the data in the XML document having the XML data shape; and

mapping the data in the XML document from the default XML representation to the data shape of the JAVA class; and

generating a runtime JAVA object containing data obtained from the XML document, said data having the data shape of the JAVA class.

27. (Previously Presented) The method of claim 26, further comprising:

combining a plurality of XML documents and scalar values in order to generate a single runtime JAVA object.

28. (Previously Presented) The method of claim 26, further comprising:

storing a plurality of default XML representations in a transformation file;

attaching the transformation file to the workflow, such that the plurality of default XML representations is referenced from within the workflow.

29. (Previously Presented) The method of claim 26, wherein an engine receives a query language specification and generates a sequence of rules used to transform the XML document from the XML data shape to the data shape of the JAVA class.

30 (Previously Presented) The method of claim 28, wherein the transformation file is included in the workflow by implementing a control within the workflow, said control invoking transformations during send or receive operations from the workflow.